IN THE CLAIMS:

Please cancel claims 4-5 and 14-15 without prejudice. Please amend the claims as follows, substituting any amended claim(s) for the corresponding pending claim(s):

Claim 1 (Canceled).

2. (Currently Amended) A pipelined microprocessor ~~capable of~~ detecting ~~an~~ a first instruction ~~that loads~~ using first base and offset address values to load data from a first memory location that was previously stored to, wherein the first instruction is detected based upon the first base and offset address values and without ~~requiring computation of an external~~ using a memory address ~~of said first memory location for the instruction~~ corresponding to the first base and offset address values.

3. (Currently Amended) A pipelined microprocessor as claimed in Claim 2 wherein ~~said~~ the pipelined microprocessor ~~is capable of detecting an~~ detects a second instruction ~~that stores~~ using second base and offset address values to store data into a second memory location that was previously read from, wherein the second instruction is detected based upon the second base and offset address values and without ~~computing an external~~ using a memory address ~~of said second memory location second memory location~~ corresponding to the second base and offset address values.

Claims 4-5 (Canceled).

6. (Currently Amended) A pipelined microprocessor as claimed in Claim 4 2 wherein ~~said~~ the pipelined microprocessor ~~is capable of examining symbolic structure of said~~ examines base and offset address values used to access memory locations by store instructions that ~~load~~ store data into the memory locations ~~from identical memory locations that were previously stored to~~, and ~~capable of detecting said~~ detects load instructions that load data from ~~identical~~ memory locations ~~by examining said symbolic structure~~ corresponding to base and offset address values identical to the base and offset address values used by the store instructions.

7. (Currently Amended) A pipelined microprocessor as claimed in Claim ~~5~~ 3 wherein ~~said~~ the pipelined microprocessor ~~is capable of examining symbolic structure of said~~ examines base and offset address values used to access memory locations by load instructions that ~~store~~ load data ~~into identical memory locations that were previously read~~ from the memory locations, and ~~capable of detecting said~~ detects store instructions that store data into ~~identical~~ memory locations ~~by examining said symbolic structure~~ corresponding to base and offset address values identical to the base and offset address values used by the load instructions.

8. (Currently Amended) A pipelined microprocessor as claimed in Claim 6 wherein ~~said~~ the pipelined microprocessor ~~is capable of detecting said instructions that load data from identical memory locations by identifying an~~ detects identical offset address values ~~from an~~ and identical base address values in at least one register within ~~said~~ the pipelined microprocessor.

9. (Currently Amended) A pipelined microprocessor as claimed in Claim 7 wherein ~~said~~ the pipelined microprocessor ~~is capable of detecting said instructions that load data from identical memory locations by identifying an~~ detects identical offset address values ~~from an~~ and identical base address values in at least one register within ~~said~~ the pipelined microprocessor.

10. (Currently Amended) A pipelined microprocessor as claimed in Claim 6 wherein ~~said~~ the pipelined microprocessor comprises:

an instruction decode stage ~~capable of~~ detecting ~~said~~ load instructions that load data from ~~identical~~ memory locations ~~by identifying an identical~~ corresponding to offset address values ~~from an identical~~ and base address values ~~in a register within said pipelined microprocessor~~ identical to offset address values and base address values used by prior store instructions that store data into the memory locations; and

a bypass element ~~capable of~~ sending a bypass signal to an instruction execution stage of ~~said~~ the pipelined microprocessor that indicates that ~~said~~ a load instructions ~~refer to an identical memory location~~ uses a base address value and an offset address value identical to a base address value and an offset address value used by a prior store instruction.

11. (Currently Amended) A pipelined microprocessor as claimed in Claim 7 wherein ~~said~~ the pipelined microprocessor comprises:

an instruction decode stage ~~capable of~~ detecting ~~said~~ store instructions that store data into ~~identical~~ memory locations ~~by identifying an identical~~ using offset address values ~~from an identical~~ and base address values ~~in a register within said pipelined microprocessor~~ identical to offset address values and base address values used by prior load instructions that load data from memory locations; and

a bypass element ~~capable of~~ sending a bypass signal to an instruction execution stage of ~~said~~ the pipelined microprocessor that indicates that ~~said~~ a store instructions ~~refer to an identical memory location~~ uses a base address value and an offset address value identical to a base address value and an offset address value used by a prior load instruction.

12. (Currently Amended) A method for operating a pipelined microprocessor, ~~said method~~ comprising:

detecting, in ~~said~~ the pipelined microprocessor, ~~an~~ a first instruction ~~that loads~~ using first base and offset address values to load data from a first memory location that was previously stored to, wherein the first instruction is detected based upon the first base and offset address values and without ~~requiring computation of an external~~ using a memory address ~~of said first memory location for the instruction~~ corresponding to the first base and offset address values.

13. (Currently Amended) A method for operating a pipelined microprocessor as claimed in Claim 12, ~~said method~~ further comprising:

detecting, in ~~said~~ the pipelined microprocessor, ~~an~~ a second instruction ~~that stores~~ using second base and offset address values to store data into a second memory location that was previously read from, wherein the second instruction is detected based upon the second base and offset address values and without ~~computing an external~~ using a memory address ~~of said second memory location~~ corresponding to the second base and offset address values.

Claims 14-15 (Canceled).

16. (Currently Amended) A method for operating a pipelined microprocessor as claimed in Claim ~~14~~ 12, ~~said method~~ further comprising:

examining, in ~~said~~ the pipelined microprocessor, ~~symbolic structure of said~~ base and offset address values used to access memory locations by store instructions that ~~load~~ store data into the ~~from identical~~ memory locations ~~that were previously stored to~~; and

detecting ~~said~~ load instructions that load data from ~~identical~~ memory locations ~~by examining said symbolic structure~~ corresponding to base and offset address values identical to the base and offset address values used by the store instructions.

17. (Currently Amended) A method for operating a pipelined microprocessor as claimed in Claim ~~15~~ 13, ~~said method~~ further comprising:

examining, in ~~said~~ the pipelined microprocessor, ~~symbolic structure of said~~ base and offset address values used to access memory locations by load instructions that ~~store~~ load data ~~into identical~~ from memory locations ~~that were previously read from~~; and

detecting said instructions that store data into ~~identical~~ memory locations ~~by examining said symbolic structure~~ corresponding to base and offset address values identical to the base and offset address values used by the load instructions.

18. (Currently Amended) A method for operating a pipelined microprocessor as claimed in Claim 16, ~~said method~~ further comprising:

detecting, in an instruction decode stage of ~~said~~ the pipelined microprocessor, ~~said~~ load instructions that load data from ~~identical~~ memory locations ~~by identifying an identical~~ corresponding to offset address values and ~~from an identical~~ base address values ~~in a register within said pipelined microprocessor~~ identical to offset address values and base address values used by prior store instructions that store data into the memory locations; and

sending a bypass signal from a bypass element to an instruction execution stage of ~~said~~ the pipelined microprocessor, wherein ~~said~~ the bypass signal indicates that a load instructions ~~refer to an identical memory location~~ uses a base address value and an offset address value identical to a base address value and an offset address value used by a prior store instruction.

19. (Currently Amended) A method for operating a pipelined microprocessor as claimed in Claim 17, ~~said method~~ further comprising:

detecting, in an instruction decode stage of ~~said~~ the pipelined microprocessor, ~~said~~ store instructions that store data into ~~identical~~ memory locations ~~by identifying an identical~~ using offset address values ~~from an identical~~ and base address values ~~in a register within said pipelined microprocessor~~ identical to offset address values and base address values used by prior load instructions that load data from memory locations; and

sending a bypass signal from a bypass element to an instruction execution stage of ~~said~~ the pipelined microprocessor, wherein ~~said~~ the bypass signal indicates that ~~said~~ a load instructions ~~refer to an identical memory location~~ uses a base address value and an offset address value identical to a base address value and an offset address value used by a prior store instruction.

20. (Currently Amended) A method for operating a pipelined microprocessor, ~~said method~~ comprising:

detecting a first instruction that stores data to a first memory location, ~~said~~ the first instruction comprising syntax for computing an effective address for ~~said~~ the first memory location;

detecting a second instruction that loads data from a second memory location, ~~said~~ the second instruction comprising syntax for computing an effective address for said second memory location;

determining ~~said~~ the syntax for ~~said~~ the first instruction and ~~said~~ the syntax for ~~said~~ the second instruction;

using ~~said~~ the syntax for ~~said~~ the first instruction and ~~said~~ the syntax for ~~said~~ the second instruction to determine a relationship between ~~said~~ the first memory location and ~~said~~ the second memory location, without ~~requiring computation of said~~ using the effective address for ~~said~~ the first memory location ~~and without requiring computation of said~~ or the effective address for ~~said~~ the second memory location; and

using ~~said~~ the relationship to determine whether to perform one of ~~said~~ the first instruction and ~~said~~ the second instruction.

21. (Currently Amended) A method for operating a pipelined microprocessor as claimed in Claim 20, wherein ~~said~~ the syntax for ~~said~~ the first instruction and ~~said~~ the syntax for ~~said~~ the second instruction refer to an identical memory location.